README file – Project 2

Phase 3

Implementation –

The APIs and technology involved for the implementation are as mentioned –

* Python using Flask for the web UI
* Docker toolbox that helps setup container using VirtualBox

Code flow –

NOTE:

* The application is transferred to the Docker with database support. You can access the UI at <http://127.0.0.1:5000>
* Always start Docker before starting localhost, you can do this by starting Docker Quickstart Terminal, once it has finished with the setup, the localhost can be started.
* The application allows a user to be both publisher and subscriber.

Directory structure for the codebase –

Steps to execute –

* Open terminal and change directory to *web*
* Run the following commands –
  + docker-compose build
  + Docker-compose up
* Open browser at [http://192.168.99.100:5000](http://192.168.99.100:50001) (docker-ip:5000)

Working application screenshots –

|  |
| --- |
|  |
| Landing page1 |
|  |
| Landing page2 |
|  |
| Landing page3 |
|  |
| Main page(User:ankita): Publisher(TV Shows)-Subscriber(Gadgets) |
|  |
| Main page(User:kushagra): Publisher(Football)-Subscriber(Gadgets, TV Shows) |
|  |
| Main page(User:anon): Publisher(Gadgets)-Subscriber(TV Shows, Football) |
|  |
| Notify Subscriber for any published data for subscribed topic |

Logs –

*ak@ak:~/Downloads/DS\_P2\_2/web$ sudo docker-compose up*

*[sudo] password for ak:*

*Creating web\_db\_1*

*Creating web\_web\_1*

*Attaching to web\_db\_1, web\_web\_1*

*db\_1 | 2018-11-13T04:23:58.131+0000 I JOURNAL [initandlisten] journal dir=/data/db/journal*

*db\_1 | 2018-11-13T04:23:58.132+0000 I JOURNAL [initandlisten] recover : no journal files present, no recovery needed*

*web\_1 | \* Serving Flask app "app" (lazy loading)*

*…*

*db\_1 | 2018-11-13T04:24:01.453+0000 I JOURNAL [initandlisten] preallocateIsFaster=true 48.12*

*…*

*db\_1 | 2018-11-13T04:24:06.712+0000 I JOURNAL [initandlisten] preallocating a journal file /data/db/journal/prealloc.0*

*web\_1 | 172.17.0.1 - - [13/Nov/2018 04:24:06] "GET / HTTP/1.1" 200 -*

*web\_1 | 172.17.0.1 - - [13/Nov/2018 04:24:14] "POST / HTTP/1.1" 200 -*

*db\_1 | 2018-11-13T04:24:19.580+0000 I JOURNAL [initandlisten] preallocating a journal file /data/db/journal/prealloc.1*

*db\_1 | 2018-11-13T04:24:33.348+0000 I JOURNAL [initandlisten] preallocating a journal file /data/db/journal/prealloc.2*

*db\_1 | 2018-11-13T04:24:46.062+0000 I JOURNAL [durability] Durability thread started*

*db\_1 | 2018-11-13T04:24:46.062+0000 I JOURNAL [journal writer] Journal writer thread started*

*db\_1 | 2018-11-13T04:24:46.104+0000 I CONTROL [initandlisten] MongoDB starting : pid=1 port=27017 dbpath=/data/db 64-bit host=b29eaaa3b386*

*db\_1 | 2018-11-13T04:24:46.104+0000 I CONTROL [initandlisten] db version v3.0.2*

*db\_1 | 2018-11-13T04:24:46.105+0000 I CONTROL [initandlisten] git version: 6201872043ecbbc0a4cc169b5482dcf385fc464f*

*db\_1 | 2018-11-13T04:24:46.105+0000 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.0.1e 11 Feb 2013*

*db\_1 | 2018-11-13T04:24:46.105+0000 I CONTROL [initandlisten] build info: Linux ip-10-171-120-232 3.2.0-4-amd64 #1 SMP Debian 3.2.46-1 x86\_64 BOOST\_LIB\_VERSION=1\_49*

*db\_1 | 2018-11-13T04:24:46.105+0000 I CONTROL [initandlisten] allocator: tcmalloc*

*db\_1 | 2018-11-13T04:24:46.105+0000 I CONTROL [initandlisten] options: {}*

*db\_1 | 2018-11-13T04:24:46.105+0000 I INDEX [initandlisten] allocating new ns file /data/db/local.ns, filling with zeroes...*

*db\_1 | 2018-11-13T04:24:46.338+0000 I STORAGE [FileAllocator] allocating new datafile /data/db/local.0, filling with zeroes...*

*db\_1 | 2018-11-13T04:24:46.338+0000 I STORAGE [FileAllocator] creating directory /data/db/\_tmp*

*db\_1 | 2018-11-13T04:24:46.373+0000 I STORAGE [FileAllocator] done allocating datafile /data/db/local.0, size: 64MB, took 0.016 secs*

*db\_1 | 2018-11-13T04:24:46.392+0000 I NETWORK [initandlisten] waiting for connections on port 27017*

*db\_1 | 2018-11-13T04:24:46.736+0000 I NETWORK [initandlisten] connection accepted from 172.17.0.3:39298 #1 (1 connection now open)*

*…*

*db\_1 | 2018-11-13T04:24:46.742+0000 I INDEX [conn2] allocating new ns file /data/db/tododb.ns, filling with zeroes...*

*db\_1 | 2018-11-13T04:24:47.006+0000 I STORAGE [FileAllocator] allocating new datafile /data/db/tododb.0, filling with zeroes...*

*db\_1 | 2018-11-13T04:24:47.058+0000 I STORAGE [FileAllocator] done allocating datafile /data/db/tododb.0, size: 64MB, took 0.051 secs*

*db\_1 | 2018-11-13T04:24:47.084+0000 I WRITE [conn2] insert tododb.pubData query: { publishedData: { Football: "asassasasa,asassasasa" }, \_id: ObjectId('5bea520e4b880f000e33c411') } ninserted:1 keyUpdates:0 writeConflicts:0 numYields:0 locks:{ Global: { acquireCount: { w: 2 } }, MMAPV1Journal: { acquireCount: { w: 8 } }, Database: { acquireCount: { w: 1, W: 1 } }, Collection: { acquireCount: { W: 1 } }, Metadata: { acquireCount: { W: 4 } } } 341ms*

*db\_1 | 2018-11-13T04:24:47.084+0000 I COMMAND [conn2] command tododb.$cmd command: insert { insert: "pubData", ordered: true, documents: [ { publishedData: { Football: "asassasasa,asassasasa" }, \_id: ObjectId('5bea520e4b880f000e33c411') } ] } keyUpdates:0 writeConflicts:0 numYields:0 reslen:40 locks:{ Global: { acquireCount: { w: 2 } }, MMAPV1Journal: { acquireCount: { w: 8 } }, Database: { acquireCount: { w: 1, W: 1 } }, Collection: { acquireCount: { W: 1 } }, Metadata: { acquireCount: { W: 4 } } } 342ms*

*db\_1 | 2018-11-13T04:24:47.084+0000 I WRITE [conn3] insert tododb.pubData query: { publishedData: { Football: "asassasasa,asassasasa" }, \_id: ObjectId('5bea520e4b880f000e33c412') } ninserted:1 keyUpdates:0 writeConflicts:0 numYields:0 locks:{ Global: { acquireCount: { w: 1 } }, MMAPV1Journal: { acquireCount: { w: 3 } }, Database: { acquireCount: { w: 1 }, acquireWaitCount: { w: 1 }, timeAcquiringMicros: { w: 341340 } }, Collection: { acquireCount: { W: 1 } } } 341ms*

*db\_1 | 2018-11-13T04:24:47.084+0000 I COMMAND [conn3] command tododb.$cmd command: insert { insert: "pubData", ordered: true, documents: [ { publishedData: { Football: "asassasasa,asassasasa" }, \_id: ObjectId('5bea520e4b880f000e33c412') } ] } keyUpdates:0 writeConflicts:0 numYields:0 reslen:40 locks:{ Global: { acquireCount: { w: 1 } }, MMAPV1Journal: { acquireCount: { w: 3 } }, Database: { acquireCount: { w: 1 }, acquireWaitCount: { w: 1 }, timeAcquiringMicros: { w: 341340 } }, Collection: { acquireCount: { W: 1 } } } 341ms*

*web\_1 | 172.17.0.1 - - [13/Nov/2018 04:24:47] "POST / HTTP/1.1" 200 -*

*web\_1 | 172.17.0.1 - - [13/Nov/2018 04:24:47] "POST / HTTP/1.1" 200 -*

*db\_1 | 2018-11-13T04:24:51.013+0000 I NETWORK [initandlisten] connection accepted from 172.17.0.3:39304 #4 (4 connections now open)*